Automatic Calibration of Large Traffic Models

Félix Mézière* Gabriel Gomes**
California PATH Program
University of California Berkeley

Abstract

Some abstract

Index Terms

Large traffic, calibration, imputation, optimization, black box, evolutionnary algorithm, freeway model, CTM.

Introduction

SOME introduction

- What problem we solve
- History of the problem
- Our method and what it contributes
- What is a large traffic model, what is it for, why calibrate it.
- Explain shortly how we are gonna calibrate, mention *knobs, templates, fitness function to minimize, performance calculators, fitting real data*
- This work is mostly empirical/experimental, it is a base.
- Announce in what order we are going to proceed ('first blablabla, then blablabla ... finally blablabla).

we should probably insist from the beginning on the congestion pattern matching, which is the main feature.

I. Modeling and notation

A. Freeway model

We consider a one-way segment of freeway and its on-ramps and off-ramps. A fraction of the links is monitored.
Define without notation:

- topography of the freeway
- scenario
- links
- lanes
- nodes
- mainline
- on-ramp
- off-ramp
- ramps
- source
- sink
- linear order
- flow
- density

*Ecole polytechnique [felix.meziere@polytechnique.edu]
**University of California Berkeley [gomes@path.berkeley.edu]

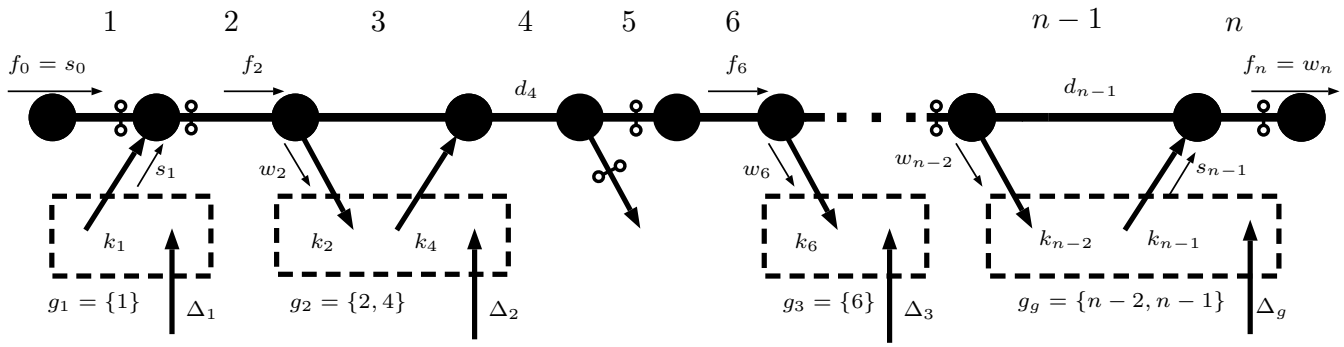


Fig. 1. Freeway model and notation

- demand
- monitored link
- other stuff that I've not thought about

Freeway model assumptions:

- the entrance link is monitored
- the behavior of the HOV
- other assumptions I've not thought about

B. Traffic model

The *large traffic model* to calibrate is characterized by the following inputs and outputs:

- *Input:*
  - duration of the scenario and time step.
  - value of the exit flow demand at every source and non-mainline sink for every time step (total over the time step).
  - other parameters proper to the model, assumed to be already calibrated
- *Output:*
  - value of the exit flow on every link, for every time step (total over the time step).
  - value of the density on every link, for every time step (average over the time step).

Traffic model assumptions:

- ≈ 0 cars beginning and end of the day
- ≈no queue on the ramps (at least on the off-ramps). Partially justified by the fact that we only input flows that are inferior to the capacity of each ramp (physical box constraints).
- other assumptions I've not thought about

C. Data

We assume that we are in possession of the following measurements on the freeway:

- Value of the exit flow on some of the mainlines links, for every time step.
- Value of the density on the same mainline links, for every time step.
- Value of the exit flow on some of the ramps, for every time step.

Data assumptions:

- The first and last mainline link (entrance and exit of the freeway portion) are monitored.

D. Notation

1) Freeway model: Let $M = [\![1, n]\!]$ the set of *mainline link* indexes and $R \subset M$ the set of mainline link indexes whose exit node is connected to a ramp (i.e. the set of ramp indexes).
We denote as $T \subset M$ the set of monitored mainline links (T for 'tracked') and $K = \{i_1, i_2, ..., i_\kappa\} \subset R$ the set of the κ non-monitored ramps (K for "knobs").
$(L_i)_{i \in M}$ are the lengths of the mainline links.

2) Traffic model: Let dt the time step of the model and D its duration. There is a total of $\frac{D}{dt} + 1$ time steps in the scenario. We will denote the set of these times steps $\tau = dt.[\![0, \frac{D}{dt}]\!]$
$\forall t \in \tau$, $(f_i(t))_{i \in M}$ and $(r_i(t))_{i \in R}$ are the flows exiting respectively the mainline and ramp links at time t, output by the model.
In addition, $f_0(t)$ is the flow entering the first mainline link (entrance of the freeway).
$\forall t \in \tau$, $(d_i(t))_{i \in M}$ are the densities on the mainline links, output by the model.
$\forall t \in \tau$, $\{\bar{f}_0(t); (\bar{r}_i(t))_{i \in M}\}$ are the exit flow demands imputed to the model.

3) Data: As a common pattern, the measured values will be denoted with a tilde.
Measured mainline exit flows are denoted $(\widetilde{f}_i(t))_{i \in T}$ Measured mainline densities are denoted $(\widetilde{d}_i(t))_{i \in T}$ Measured ramp exit flows are denoted $(\widetilde{r}_i(t))_{i \in (R \setminus K)}$

Fig. 1 summarizes the model. The missing notation will be introduced progressively.

II. Uncertainty

Our problem involves three sources of uncertainty:

- Uncertainty on the data: the measurements have a certain confidence interval.
- Uncertainty due to the inexactness of the model itself.
  This uncertainty reflects the fact that, even if we had perfect data and the demand on every link, the model would not output the exact real traffic (and congestion phenomena etc.).
- Uncertainty due to the inexact shape of the templates.

<- We gotta decide how much we develop this
These uncertaintes are merged into two uncertainties:

- *Uncertainty on the local duration-long measurements:* This describes the uncertainty at the link level. It is applied to the sum of the flow measurements of one sensor for all the duration.
  Denoting $F_i = \sum_{t \in \tau} f_i(t)$ and $\widetilde{F_i} = \sum_{t \in \tau} \widetilde{f}_i(t)$, this local uncertainty is divided into two competing components:
  - *additive local uncertainty*: denoted U^{add}. The additive confidence interval for F_i is:

    $$F_i \in [\widetilde{F_i} - U^{add},\ \widetilde{F_i} + U^{add}]$$
  - *multiplicative local uncertainty*: denoted U^{mul}. The multiplicative confidence interval for F_i is:

    $$F_i \in [\widetilde{F_i}.(1 - U^{mul}),\ \widetilde{F_i}.(1 + U^{mul})]$$
- *Uncertainty on the global duration-long measurements*: This describes the uncertainty at the whole mainline level. It is a generic multiplicative uncertainty applied to all quantities that are computed from the measurements on every mainline sensor during the whole duration.
  We denote this uncertainty U^{global}. Let $\widetilde{q_i}(t)$ a quantity computed from the measurements on link i at time t. Denoting $Q = \sum_{i \in T} \sum_{t \in \tau} q_i(t)$ and $\widetilde{Q} = \sum_{i \in T} \sum_{t \in \tau} \widetilde{q_i}(t)$, the global confidence interval for this quantity is:

  $$Q \in [\widetilde{Q}.(1 - U^{global}),\ \widetilde{Q}.(1 + U^{global})]$$

<- We've got to decide how much we justify this definition of the uncertainties

III. Problem formulation

A. Introduction

For every monitored source or sink (except the mainline sink), we input the measured flow to the model as exit flow demand.
The assumptions made in I-B imply that this demand is approximately equal to the actual flow going through the ramp, for all times and above mentioned links:

$$\forall i \in R \backslash K,\ \forall t \in \tau,\ \widetilde{r_i}(t) = \bar{r_i}(t) \approx r_i(t) \tag{1}$$

Therefore, the only missing parameters to the model are the flow demand profiles of the non monitored ramps: $(\bar{r_i}(t))_{\substack{i \in K \\ t \in \tau}}$. Our method consists in mapping these κ flow profiles into one parameter each.
To do that, a flow profile called *template* is built for every non-monitored ramp. These templates, denoted $(t_i(t))_{\substack{i \in K \\ t \in \tau}}$, consist in a normalized flow profile: a flow value is given to each element of τ and the resulting profile is normalized to a reasonable value T. For each of the non-monitored ramps i, we define a multiplicative factor k_i called *knob* that will set the intensity of the template. That is, we input as exit flow demand of the ramp its corresponding template multiplied by the ramp knob : $k_i.t_i(t)$.

The parameters of our imputation problem are therefore the κ **knobs**, corresponding to the κ non-monitored ramps.

In addition, due the same assumptions that gave Eq. 1, we have :

$\forall i \in K,\ \forall t \in \tau,\ k_i.t_i(t) = \bar{r_i}(t) \approx r_i(t)$
and, especially, with $\Theta = \sum_{t \in \tau} t_i(t)$:

$$\sum_{t \in \tau} r_i(t) = k_i.\Theta \tag{2}$$

B. Constraints on the parameters

We define here the constraints verified by the knobs. They consist in box hard boundaries and linear inequalities.
Notation:

$\vec{k} = (k_{i_1}, k_{i_2}, ..., k_{i\kappa})$ is the vector containing the values of the knobs.
$\sigma = (\sigma_{i_1}, \sigma_{i_2}, ..., \sigma_{i_\kappa})$ is the source/sink indicator vector for the knobs:

$$\forall j \in K,\ \sigma_j = \begin{cases} 1\ if\ ramp\ j\ is\ an\ on\text{-}ramp \\ -1\ if\ ramp\ j\ is\ an\ off\text{-}ramp \end{cases}$$

For clarity, we will often abusively use the expression *knob-ramp i* instead of *ramp corresponding to the knob i*.

1) Physical boundaries: The box constraints applied to each knob are physical capacity limits imposed by the ramp they are associated with. They reflect that the maximum value of the ramp flow cannot exceed the capacity of the ramp.

$\forall i \in K$, the maximum m_i of knob i is defined by:

$\forall t \in \tau$,
$k_i.t_i(t) = \bar{r_i}(t) \leq$ [*Capacity of the ramp associated to knob i*]

$$\Rightarrow m_i = \frac{[Capacity\ of\ the\ ramp\ associated\ to\ knob\ i]}{\max_t t_i(t)}$$

We impose therefore:

$\forall i \in K, \forall p \in \mathbb{N},\ 0 \leq k_i \leq m_i$

which is equivalent to :

$$\vec{k} \in \mathscr{B} \tag{3}$$

with $\mathscr{B}$ the hyper-cube ("box") defined by :

$$\mathscr{B} = \{ \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_\kappa \end{bmatrix} \in \mathbb{R}^\kappa | \ \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leq \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_\kappa \end{bmatrix} \leq \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_\kappa \end{bmatrix} \}$$

The box $\mathscr{B}$ will be called the *search space* of the knobs.

2) Knob groups and flow balance: We define here objects and notation to describe a simple situation: the knobs are closely monitored by nearby mainline sensors, leading often to a situation where a ramp is the only non-monitored ramp between two mainline sensors. This is equivalent to it being monitored, if it was not for the uncertainties.

We call *segment* the set of links between two consecutive mainline sensors, including the links containing these sensors.
We call *knob group* each set of knobs whose corresponding ramp is connected to the same monitored segment. This definition is illustrated in Fig. 1.
We call *partially monitored segment* the monitored segments associated with a knob group i.e. containing at least one non-monitored ramp.

Denoting γ the total number of knob groups, we have:

Partially monitored segments : $(S_i)_{i \in [\![1,\gamma]\!]}$

Formal definition:
$\exists\ !\ \gamma \in M,\ \exists\ !\ ((\beta_i, \eta_i))_{i \in [\![1,\gamma]\!]} \in (T^2)^{[\![1,\gamma]\!]}$ *s.t.*,
denoting $S_i = [\![\beta_i, \eta_i]\!]$ *and* $S = \bigcup_{i \in [\![1,\gamma]\!]} S_i$:
$\forall i \in [\![1, \gamma]\!]$,
1) $[\![\beta_i, \eta_i]\!] \cap T = \{\beta_i, \eta_i\}$
2) $[\![\beta_i, \eta_i]\!] \cap K \neq \{\emptyset\}$
and $\forall k \in K,\ k \in S$.

Knob groups : $(g_i)_{i \in [\![1,\gamma]\!]}$

Formal definition:
$\forall i \in [\![1, \gamma]\!],\ g_i = S_i \cap K$

We can deduce the value of the daily flow brought by the knobs of each group from the *knob group flow balance*: the difference between all the flows entering and all the flows exiting their incomplete monitored segment. That is the sum of the flow exiting the mainline entrance of the segment and the flows exiting the monitored on-ramps throughout the segment minus the sum of the flow exiting the mainline exit of the

segment and the flows exiting the monitored off-ramps throughout the segment.
Put here the paragraph Gabriel wrote

Knob group flow balances : $(\Delta_i)_{i\in[\![1,\gamma]\!]}$

Formal definition:
$\forall i \in [\![1,\gamma]\!],\ \forall t \in \tau,$
$\Delta_i = \sum_{t\in\tau}\left[\widetilde{f}_{\beta_i}(t) - \widetilde{f}_{\eta_i}(t) + \sum_{j\in(R\setminus K)\cap S_i} \sigma_j.\widetilde{r_j}(t)\right]$

The balance equation of each partially monitored segment is:

$$0 = \sum_{t\in\tau}\left[f_{\beta_i}(t) - f_{\eta_i}(t) + \sum_{j\in R\cap S_i} \sigma_j.r_j(t)\right]$$

As stated in eq. 1, the model-output flows exiting the ramps are equal to their demand flows. The balance equation becomes therefore:

$$0 = \Delta_i + \sum_{t\in\tau}\left[\sum_{j\in g_i} \sigma_j.r_j(t)\right]$$

Leading to:

$$\Delta_i = -\sum_{t\in\tau}\left[\sum_{j\in g_i} \sigma_j.r_j(t)\right]$$
$$\Leftrightarrow \Delta_i = -\sum_{j\in g_i}\sigma_j\left[\sum_{t\in\tau} r_j(t)\right]$$

and thanks to eq. 2:

$$\Leftrightarrow \quad \Delta_i = -\sum_{j\in g_i} \sigma_j.k_j.\Theta \tag{4}$$

Eq. 4 shows that, for every knob group, the knobs composing it are linked by one linear equation.
This equation determines uniquely the value of the single-knob groups and links the multiple-knob groups with one linear constraint. The next paragraph describes how we apply uncertainties to this equation in order to produce new, closer to reality knob boundaries.

3) Refined knob boundaries: The local uncertainty described in II prevents us from keeping Eq. 4 as a constraint for the parameters.

$\forall i \in [\![1,\gamma]\!],$ *let the most permissive uncertainties* :
$\Delta_i^- = \max\{|\Delta_i| - U^{add}; |\Delta_i|.(1-U^{mul})\}$
$\Delta_i^+ = \max\{|\Delta_i| + U^{add}; |\Delta_i|.(1+U^{mul})\}$

Taking the local uncertainty into account in Eq. 4 is translated into the following linear inequality constraint:

$$\forall i \in [\![1,\gamma]\!],\ \Delta_i^- \leq |\sum_{j\in g_i} \sigma_j.k_j.\Theta| \leq \Delta_i^+ \tag{5}$$

These γ inequalities drastically reduce the size of the search space, defining a new *feasible space*.

Comments:
We can now illustrate and justify the form that we have adopted for the uncertainty. This form allows us to quantify the freedom given to the result: the flow balance of each knob-group is between $(1-U^{mul})$ and $(1+U^{mul})$ times what has been measured by the mainline sensors, aknowledging that we don't accept less

than $\pm U^{add}$ cars precision on the measures.
Taking into account U^{add} is indispensable. This is observed in the case of single-knob groups, where Eq. 4 leads to a unique value for the knob of the group. Let us call it *perfect value of the knob i*, denoted k_i^*. It immediately follows from Eq. 5 that two new boundaries are set for k_i, if they are tighter than $[0, m_i]$. If the perfect value is a ridiculously small quantities, the maximum obtained with $(1 \pm U^{mul}).k_i^*$ corresponds often to a total daily flow of less than 50 cars exiting the ramp, which is not acceptable.

Example: One of the ramps has a perfect value of 0.02, which leads to a maximum of ?? cars going through the ramp during the whole day if U^{mul} is set to 100% (very permissive: the daily flow can double what is measured by the mainline sensors). This is too small for any scenario.
The fluctuation allowed by the new boundaries of this ramp is of ?? *cars*, which is ?? times smaller than U^{add}: the sensors do not have this level of precision, and the sensor noise/bias is responsible for this impossible perfect value.
Once U^{add}=[10%.(*measured daily mainline flow*)] is taken into account, the maximum of the knob becomes 0.7, which corresponds to ?? cars and offers an acceptable range to flow through the ramp.

C. Performance metrics and error calculators

Three performance metrics are used on the model output to measure the state of the freeway. Each of these metrics is then compared by an error calculator to its value computed on the data. The errors are expressed as a percentage, in order to monitor easily their relative importance. As exposed just below, the common principle for this error computation is to calculate the relative difference between the model output performance and the data performance.

1) Vehicle Hours Travelled (VHT): This quantity is the sum of the time spent on the mainline by each car, over the whole duration. Obviously, it is computed using only the monitored mainline links, for the comparison with the data to be relevant. explain why it is important in traffic study, why we have chosen it

VHT computation on monitored mainline links output and data:

$$VHT(\vec{k}) = \frac{dt}{[1\ hour]} \sum_{i \in T} L_i \sum_{t \in \tau} d_i(t)$$

Denoting $\widetilde{VHT}$ the value computed from the data using the same formula, the error is the relative difference :

$$E_{VHT}(\vec{k}) = \frac{|VHT(\vec{k}) - \widetilde{VHT}|}{\widetilde{VHT}}$$

2) Vehicle Miles Travelled (VMT): This quantity is the sum of the distance traveled on the mainline by each car, over the whole duration. Obviously, it is computed using only the monitored mainline links, for the comparison with the data to be relevant. Explain why it is important in traffic study, why we have chosen it
VMT computation on monitored mainline links output and data:

$$VMT(\vec{k}) = \sum_{i \in T} L_i \sum_{t \in \tau} f_i(t)$$

Denoting $\widetilde{VMT}$ the value computed from the data using the same formula, the error is the relative difference :

$$E_{VMT}(\vec{k}) = \frac{|VMT(\vec{k}) - \widetilde{VMT}|}{\widetilde{VMT}}$$

Reduction of the feasible space: we present here a method used to reduce the feasible space size by forcing the knobs to match the correct VMT value.
VMT is the result of a simple *a priori* calculation that does not need the traffic model output calculation, if we know the boundary conditions at $t = 0$ and $t = D$. As exposed in I-B, we assume that these conditions are 0 *cars* on every link (D has to be big enough for these conditions to be very small in comparison with the total number of vehicles during D).

Let $VMT^{ref} = VMT(\vec{k}^{ref})$, a certain VMT reference value output by the model. We suppose that $\vec{k}^{ref}$ is some feasible knobs vector (in our case, we used $\vec{k}^{ref} = (1, ..., 1)$).
Denoting $VMT^{apriori}(\vec{k})$ the expected VMT value computed from $\vec{k}$:

$$VMT^{a\ priori}(\vec{k}) = VMT^{ref} + \sum_{i \in K} \left[\sigma_i . k_i . \Theta . \sum_{\substack{j \in T \\ j > i}} L_j \right] \tag{6}$$

The *a-priori* calculation above relies on anticipating the changes from VMT^{ref} caused by changing the knobs from $\vec{k^{ref}}$ to $\vec{k}$. For each knob, the flow change resulting from its modification is multiplied by the remaining mainline length and the "on/off-ramp indicator". All these contributions are then summed.

The linear equation 6 empowers us to constrain the input $\vec{k}$ in order to ensure $\widetilde{VMT} = VMT^{a\ priori}(\vec{k}) \approx VMT(\vec{k})$, thus reducing the size of the feasible space by one dimension:

$$\sum_{i \in K} \left[\sigma_i . k_i . \Theta . \sum_{\substack{j \in T \\ j > i}} L_j \right] + VMT^{ref} = \widetilde{VMT}$$

However, the global uncertainty applied on VMT (which is a global quantity computed from the sum over the whole time and space) forces us to loosen this constraint equation.
Denoting $\widetilde{VMT}^{-} = \widetilde{VMT}.(1 - U^{global})$ and
$\widetilde{VMT}^{+} = \widetilde{VMT}.(1 + U^{global})$, it becomes:

$$\widetilde{VMT}^{-} \leq \sum_{i \in K} \left[\sigma_i . k_i . \Theta . \sum_{\substack{j \in T \\ j > i}} L_j \right] + VMT^{ref} \leq \widetilde{VMT}^{+} \tag{7}$$

3) Congestion Pattern: We call *contour plot* the graph representing the value of a quantity on every mainline link at all times: the mainline links as absciss and time steps as ordinates. Fig. III-C3. below is an example of a density contour plot for one day on a 135 links freeway, from a traffic model output.
This plot is used to monitor easily where and when the congestion is : here, we see empirically that it is contained in the two framed parts.
In what follows, by analogy, we will call *contour domain* the set $\mathscr{P}=\{(i,t)|\ i \in M,\ t \in \tau\}$ and *pixel* each of its elements.
On the model output, we define the *congested* pixels as the ones where the density exceeds some *critical density* deduced for each link from the freeway and traffic models. The main feature of our calibration method is to fit the locations and times of these congested pixels to what the measurements indicate.
For each mainline link $i \in M$, we denote d_i^* the critical density.
We define the *output congested domain* $\mathscr{C} \in \mathscr{P}$ containing the congested pixels :

$$\mathscr{C} = \{(i,t) \in \mathscr{P} \mid d_i(t) \geq d_i^*\}$$

To define an error based on $\mathscr{C}$, a domain supposed to contain the congestion as to be determined. From the data *density contour plot* (partial, obtained only on the monitored links), we define a domain $\widetilde{\mathscr{C}} \subset \mathscr{P}$ fitting the congested pixels as best as it can following some criteria (how this domain is built depends on the amount of data the operator possesses and on his goals. In our case, $\widetilde{\mathscr{C}}$ was a set of rectangles containing all the congestion seen in the data contour plot).

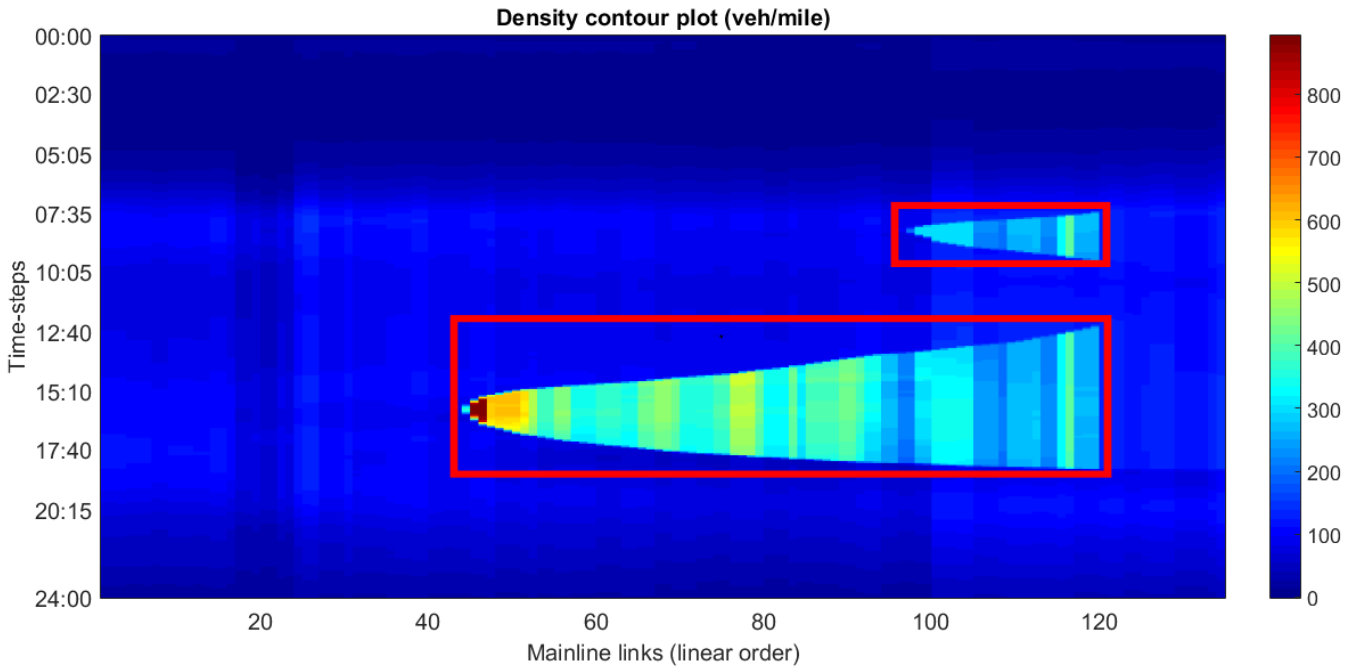


Fig. 2. Example of a density contour plot on a 135 mainline links freeway over 24h out of a traffic model output.

We can now define the congestion pattern error denoted E_{CP} as the normalized number of wrong congestion state pixels. That is, we add one to E_{CP} for each pixel that is not congested but should and for each pixel that is congested but shouldn't. We then divide this result by the number of pixels that should be congested (i.e. $Card(\widetilde{\mathscr{C}}) = \sum_{t\in\tau}\sum_{i\in M}\mathbb{1}_{(i,t)\in\widetilde{\mathscr{C}}}$).

$$E_{CP}(\vec{k}) = \frac{\sum_{t\in\tau}\sum_{i\in M}\mathbb{1}_{\{(i,t)\in(\widetilde{\mathscr{C}}\setminus\mathscr{C})\cup(\mathscr{C}\setminus\widetilde{\mathscr{C}})\}}}{\sum_{t\in\tau}\sum_{i\in M}\mathbb{1}_{(i,t)\in\widetilde{\mathscr{C}}}}$$

Fig. 3. below is a visual representation on a contour domain of $(\widetilde{\mathscr{C}}\setminus\mathscr{C})$ otherwise called *false negative pixels*, $(\mathscr{C}\setminus\widetilde{\mathscr{C}})$ otherwise called *false negative pixels* and $(\mathscr{C}\cap\widetilde{\mathscr{C}})\cup(\mathscr{P}\setminus(\mathscr{C}\cup\widetilde{\mathscr{C}}))$, which are the *correct congestion matching* pixels. Note that E_{CP} becomes very sensible if the data does not contain much congestion (i.e. $Card(\widetilde{\mathscr{C}}) << Card(\mathscr{P})$).

D. Objective function

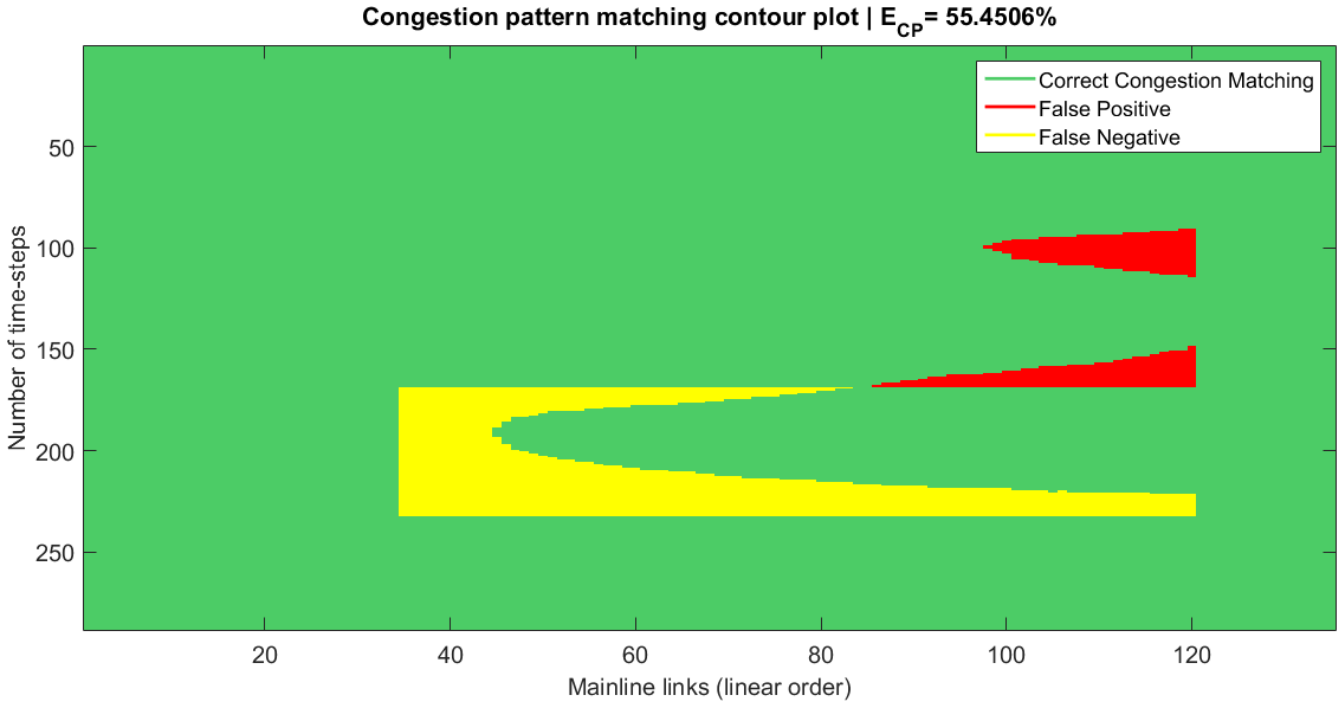
The calibration method consists in minimizing jointly the three errors described in III-C. We accomplish this goal by minimizing an *objective function* Φ, which is the weighted sum of the errors modified by the global uncertainty, as explained below.

Uncertainty handling : I^{global}, described in I-C, defines a tolerance threshold for the error results. The error results below I^{global} are set to zero, in order to avoid any discrimination between them (we do not have a level of precision below I^{global}).
Each of the three errors E will therefore be multiplied by $\mathbb{1}_{(E>I^{global})}$.

Let $(w_i)_{i\in[\![1,3]\!]}$ the weights, verifying :
$w_1 + w_2 + w_3 = 100$ *and* $\forall i \in \{1,2,3\},\ w_i \geq 0$.

Let the three error *contributions*:
$\phi_{VHT}(\vec{k}) = w_1.E_{VHT}(\vec{k}).\mathbb{1}_{(E_{VHT}>I^{global})}$

Fig. 3. Visual example of the congestion pattern matching computation.



$\phi_{VMT}(\vec{k}) = w_2.E_{VMT}(\vec{k}).\mathbb{1}_{(E_{VMT}>I^{global})}$
$\phi_{CP}(\vec{k}) = w_3.E_{CP}(\vec{k}).\mathbb{1}_{(E_{CP}>I^{global})}$

Φ is defined by:

$$\Phi : \left| \begin{array}{ccc} \mathscr{B} & \longrightarrow & [0,100] \\ \vec{k} & \longmapsto & \phi_{VHT}(\vec{k}) + \phi_{VMT}(\vec{k}) + \phi_{CP}(\vec{k}) \end{array} \right.$$

Note : the errors can lead to values superior to 1 thus giving values of Φ superior to 100% but, to simplify, we will ignore these cases that are very far from the objective.

This definition as a weighted sum of percentages implies that the values of Φ can be interpreted as a *global error percentage*. Thanks to the normalization of the errors, the weight given to each component is equivalent to the importance the operator wants to give to each one of them.

E. Optimization problem statement

The optimization problem that we solve can now be stated :

$$\begin{aligned} & minimize \quad \Phi(\vec{k}) \\ & s.t. \quad \forall i \in [\![1,\gamma]\!],\ \Delta_i^- \leq |\sum_{j \in g_i} \sigma_j.k_j.\Theta| \leq \Delta_i^+ \\ & and\ \widetilde{VMT}^- \leq \sum_{i \in K} \left[\sigma_i.k_i.\Theta.\sum_{\substack{j \in T \\ j>i}} L_j \right] + VMT^{ref} \leq \widetilde{VMT}^+ \\ & and\ \vec{k} \in \mathscr{B} \end{aligned} \tag{8}$$

IV. Numerical method

A. Requirements

The performance errors are irregular functions. In particular, the congestion pattern fitting reflects congestion phenomena. These present numerous thresholds in their non-smooth behavior.
We can also point out that these errors aren't always correlated.
Furthermore, each evaluation of the error function requires the execution of a simulation (around 5 seconds on a desktop computer), and this evaluation is the only thing accessible of Φ: there is no way of quickly computing its value or its gradient .
Therefore, the objective function is a black box.
We deduce from these observations that convex optimization methods and derivative-based methods are not adapted to our case.
The search space is a continuous hyper-cube, as explained in III-B1.
We can conclude that we study a non-linear, non-convex black-box imputation problem in continuous domain.

In addition, the resolution method has to be adaptive, since it will be applied to many different freeways, times and sensor densities. We want as few numerical method parameters to tune as possible and no prior optimization knowledge required if possible.

Finally, this experiment is a proof of concept that does not take execution time as a criteria : the goal is to obtain the best possible result quality and uniqueness (global minimum of Φ).

B. Co-variance Matrix Adaptation - Evolution Strategy (CMA-ES)

The state-of-the-art evolutionnary algorithm CMA-ES is very well suited for these requirements.
The description of its architect, Nicolaus Hansen insert bibliography here(see https://www.lri.fr/ hansen/cmaesintro.html) contains these words :

- It is conceived to solve "difficult non-linear non-convex black-box optimisation problems in continuous domain".
- It is feasible on "non-smooth and even non-continuous problems, as well as on multimodal and/or noisy problems".
- "The CMA-ES does not use or approximate gradients and does not even presume or require their existence"
- It is "competitive for global optimization".

In addition, it is extremely adaptive as only an initial standard deviation and the population size have to be tuned.
More information on CMA-ES can be found at https://www.lri.fr/ hansen/cmatutorial.pdf, especially in the parts *0.3: Randomized Black-Box optimization* and *5: Discussion.*
For further understanding, the reader can keep in memory that the algorithm samples a population Π_p of λ random points at iteration p. It then evaluates each one of them, and modifies its internal parameters so that the next λ sampled points Π_{p+1} will be more probably in the direction of the points of Π_p that gave the smaller Φ values. It globally keeps memory of the fitness (objective function value) of the points it encountered.

In what follows, the objective function Φ will also be called *fitness function.*

C. Constraints implementation

The CMA-ES source code handles box constraints natively using the "repair and penalize" procedure. This consists in repairing non-feasible sampled points before inputing them to the model; and penalizing (i.e. increasing the value of the fitness function) proportionally to the distance between the unfeasible point and the feasible space. This method forces the algorithm to eventually enter and stay in the feasible domain while avoiding evaluating non-allowed points.
However, the source code does not handle linear constraints. We describe here how we apply manually this repair & penalize procedure to reflect the two linear constraints shown in III-E.

Repairing:

Let $\vec{\underline{k}}^{(p)}$ the knobs vector sampled by CMA-ES at iteration p (i.e. before repairment).
The projection is implemented using the following program in a standard quadratic optimization solver:

$$
\begin{aligned}
&minimize \quad \left\| \vec{k}^{(p)} - \vec{\underline{k}}^{(p)} \right\|_2 \\
&s.t. \quad \forall i \in G, \quad \Delta_i^- < \textstyle\sum_{j \in g_i} \sigma_j.k_j.\Theta < \Delta_i^+ \\
&and \ \widetilde{VMT}^- \leq \sum_{i \in K} \left[\sigma_i.k_i.\Theta. \sum_{\substack{j \in T \\ j>i}} L_j \right] + VMT^{ref} \leq \widetilde{VMT}^+ \\
&and \quad \vec{k}^{(p)} \in \mathscr{B}
\end{aligned}
$$

Let us illustrate the effect of this program with an example. We suppose that the $\widetilde{TVM}^{\pm}$ condition above is enough loose for the condition to be respected without influencing the projection (often verified in practice). Fig. 4 above displays, for a two-knobs group $g_l = \{i, j\}$, the projection of several sampled points due to the

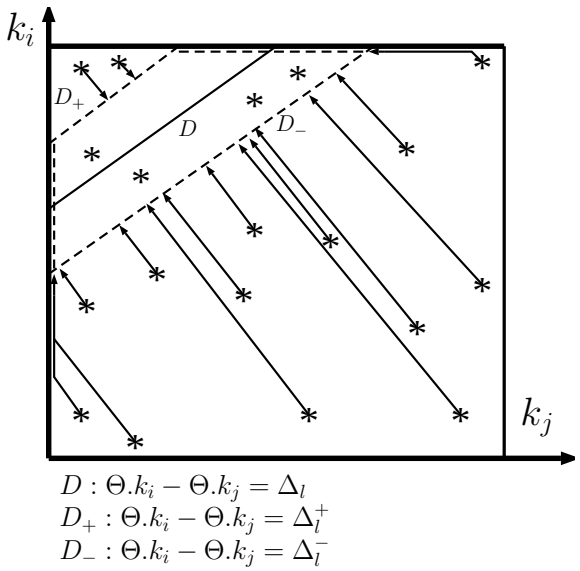


Fig. 4. Example of projection on a two-knobs group

two remaining conditions ($\Delta_l^{\pm}$ and $\mathscr{B}$). D is the hyper-plan (straight line in dimension 2) on which the knob group flow will have the exact value Δ_l. The two *tolerance hyper-plans* on which the knob group flow value will be Δ_l^+ and Δ_l^- are respectively D^+ and D^-. In addition, the external square is the hyper-cube corresponding to the physical boundaries. The points sampled in the feasible space (inside the doted line trapezium) remain untouched while the others are projected on the nearest point of the edge of the feasible space.
This repair ensures that only feasible values of the input are tested and that the algorithm will not get stuck in an unfeasible well of the physical boundaries hyper-cube.

Penalizing:

At each evaluation, a penalization proportional to the distance between the projected and original point is added to the fitness function. This ensures that the algorithm will come closer to the feasible space at every iteration until eventually entering and staying inside it. This feature is important as it avoids two imbalances on the sampling:

- Testing more points on the edges than we should: if the algorithm is left sampling points far from the edges, without penalization, it will have no incentive to prefer sampling next to the feasible space than far.

This is an obstacle from entering the feasible space, as all the points on the straight lines perpendicular to the edges would be equivalent. This would lead to a situation where it is common that too much (or all) of the points are sampled on the edges, CMA-ES converging far from the feasible space.(<- rewrite this).

- Imbalance between the edge points tested: the edge points which are the projection of more unfeasible hyper-cube points than others will be sampled unfairly more often.
  Example: in Fig. 4, the points on the mediator of the segment formed by the intersections of D and the square while inside the square are more numerous than the points on any other straight with same slope. The points on one of these straights and below the feasible space are all projected to the same point of the edge of the feasible space. Therefore, without penalization, the middle of the segment cited above would be sampled more often than the other points of its edge, for a reason that is not the simulation output it leads to.

The penalization is normalized by a factor which is the distance between the physical maximums and minimums vectors, reflecting the *order of magnitude* of the search space.

$$E_{proj}(\underline{\vec{k}}^{(p)}, \vec{k}^{(p)}) = \frac{\left\| \vec{k}^{(p)} - \underline{\vec{k}}^{(p)} \right\|_2}{\left\| \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_\kappa \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\|_2}$$

A fourth contribution $w_4.E_{proj}(\underline{\vec{k}}^{(p)}, \vec{k}^{(p)})$ is therefore added to the fitness function.

With the same definitions as in III-D and
$\sum_{i=1}^{4} w_i = 1$ *and* $\forall i \in [\![1,4]\!]\ w_i \geq 0$,

the final fitness function used is J defined by :

$$J : \left| \begin{array}{ccc} \mathscr{B} & \longrightarrow & [0,100] \\ \underline{\vec{k}}^{(p)} & \longmapsto & \Phi(\vec{k}^{(p)}) + w_4.E_{proj}(\underline{\vec{k}}^{(p)}, \vec{k}^{(p)}) \end{array} \right.$$

Single-knob group specificity: For a single-knob group, the condition Eq. 5 is equivalent to hard boundaries for the concerned knob (one equation in one dimension). In this case, the projection due to Eq. 5 is not implemented but the physical boundaries (from $\mathscr{B}$) of the knob input to CMA-ES are replaced by these new boundaries, if they are narrower :

$$\forall\ i \in G\ s.t.\ Card(g_i) = 1,\ i.e.\ g_i = \{j\}\ :$$
$$\frac{\max\{0; |\Delta_i^-|\}}{\Theta} \leq k_j \leq \frac{\min\{m_i; |\Delta_i^+|\}}{\Theta} \tag{9}$$

V. Experiment settings

We present here the elements we chose to run the experiment we made.

A. Context: Origin of the data

The real scenario used for this study is a portion of freeway 210 East in the suburbs of Los Angeles. The measurements used are collected by the sensors network PeMS of Caltrans. For more information, see http://pems.dot.ca.gov/
These daily measurements are flow, density and speed on 74 links of 188, every 5 minutes for 24 hours (289

time-steps).
After deleting partial or too biased data, the sensors have the following distribution:

- 33/135 monitored mainline links
- 26/28 monitored on-ramps
- 15/25 monitored off-ramps

This makes a total of **12 knobs**.

We chose to calibrate the model on the average of 5 Tuesdays (0am-12pm) in fall 2014 data. The goal is to find the set of knobs that best fits the profile of a Tuesday (the general shape of the traffic depending greatly on the day of the week).

For each knob-ramp, the template is built by taking the average of the two closest monitored ramps that have the same size and incoming traffic context.

B. The traffic model : CTM

We use the Cell Transmission Model, a popular model for macroscopic traffic prediction by Carlos Daganzo insert bibliography here. It consists in solving the kinematic wave equation.
CTM defines the notion of *Fundamental Diagram* for every link, that contains all the traffic properties of the link in 3 parameters : capacity, congestion speed and free-flow speed. The theoretical density congestion threshold of each link is defined by $\frac{Link\ capacity}{Link\ freeflow\ speed}$.

C. The large traffic simulator : BeATS

The macroscopic traffic simulator used is the Berkeley Advanced Traffic Simulator (BeATS). It computes the results of CTM in a mode where the entry, on-ramp AND off-ramp demands are inputs (the *split-ratios* are therefore outputs). The inputs of the simulator are:

- Freeway model.
- Fundamental Diagram of every link.
- Exit flow demand of every source and off-ramp. Time-step is 5 minutes, as for the data.

The outputs are the entry and exit flows, density and speed in every link, every 5 minutes, for the whole day.

D. Implementation of the congestion pattern

The theoretical congestion threshold d_i^* is not satisfactory as it monitors "noisy" congestion (situations of free-flow that is at the limit of the congestion), distorting E_{CP} results. We add a small number of vehicles δ (in our case, $\delta = 6$) to this threshold in order to monitor only the hardcore congestion that our congested domain $\widetilde{\mathscr{C}}$ is supposed to capture.

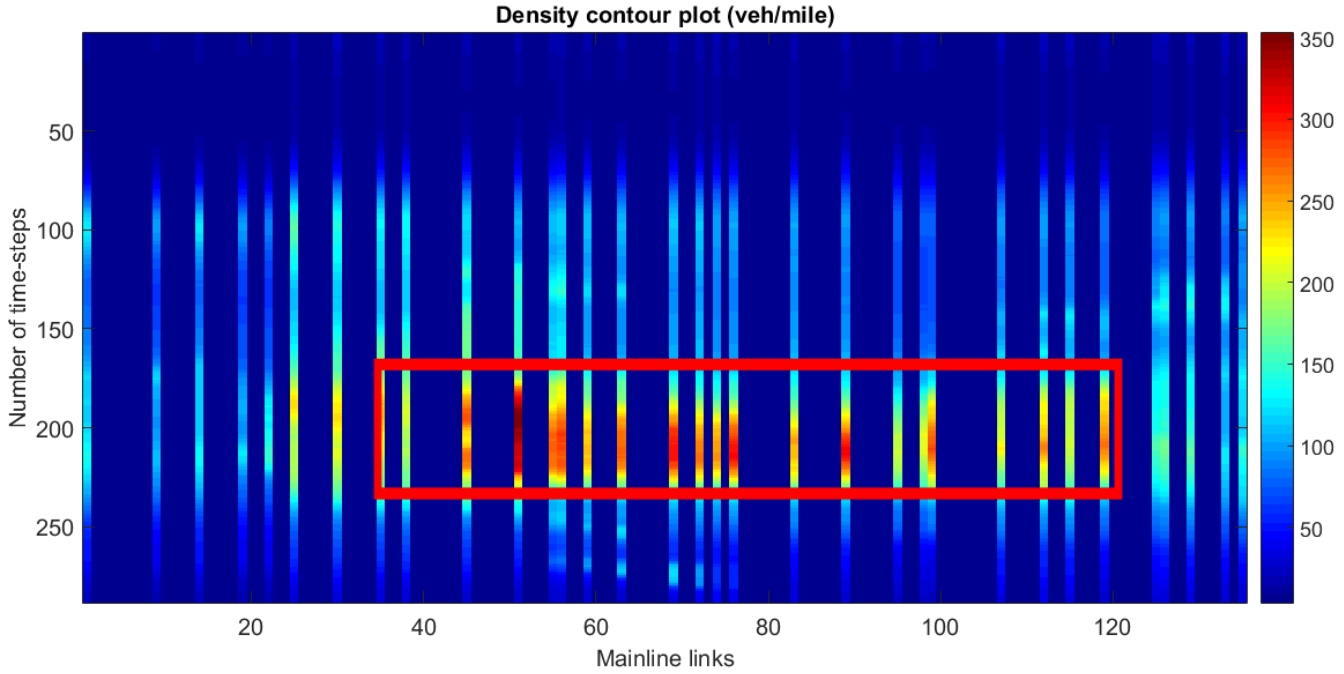
$$d_i^* = \frac{Link\ capacity}{Link\ freeflow\ speed} + \delta$$

As we are trying to fit an average congestion, we define the data congested domain $\widetilde{\mathscr{C}}$ as a set of boxes that roughly capture each congestion structure.
The average data density contour plot and the unique box we defined in our case are displayed below :

VI. Experiment results

1) Effects of changing the parameters(initial standard deviation, changing the boundaries, changing the weights, changing the uncertainty, changing size of population).
2) Describe quality and usefulness of the result
3) Talk about uniqueness. Way to improve (test experiment): increasing population size.*Should we contact the creator of CMAES to ask him about the uniqueness of the solution (i.e. how to improve it ?*
4) Is our problem "noisy" ?
5) Talk about how cmaes behaves the way we want

Fig. 5. Density contour plot on the average data from 5 Tuesdays.



6) Talk about what happens when we tune also the monitored ramps knobs.
7) Talk about issues:
   a) Limit to result quality due to templates/FDS
   b) Constraints handling has to be improved because several knobs end on their boundaries values
   c) Uncertainties are symmetric: making them fit the sensors bias (e.g. : if they always under estimate) would be better.
8) BLABLABLA

VII. Conclusion

- Multi-objective CMAES
- Find the knobs values that are a common good value for each day instead of the best for the average day
- Talk about tuning the FDS and Templates
- Talk about the uncertainty handling by Hansen and Niederberger,
- open to other stuff
- deepen understanding of "parameter sensitivity" and the adequation of having same scale for all knobs (and fds).
- BLABLA

Acknowledgment

Appendix

Model figure